

CLAIMS

What is claimed is:

1. A byte code modification method, comprising:

modifying a method's byte code instructions, said method comprising an entry point and an exit point, said modifying comprising inserting first and second additional byte code instructions into said method's byte code instructions, said first additional byte code instruction to cause a first output function to be executed for said method as a consequence of said entry point being reached during runtime, said second additional byte code instruction to cause a second output function to be executed for said method as a consequence of said exit point being reached during runtime.

2. The byte code modification method of claim 1 wherein said first output function records a time of said entry point being reached.

3. The byte code modification method of claim 2 wherein said first output function records input parameters provided to said method.

4. The byte code modification method of claim 1 wherein said first output function records input parameters provided to said method.

5. The byte code modification method of claim 1 wherein said second output function records a time of said exit point being reached.
6. The byte code modification method of claim 5 wherein said second output function records output parameters provided by said method.
7. The byte code modification method of claim 1 wherein said second output function records output parameters provided by said method.
8. The byte code modification method of claim 1 wherein said first output function increments a counter maintained for said method.
9. The byte code modification method of claim 1 wherein said second output function increments a counter maintained for said method.
10. The byte code modification method of claim 1 further comprising compiling source code prior to said modifying to produce said method's unmodified byte code instructions.
11. The byte code modification method of claim 1 wherein said byte code instructions are capable of being interpreted by a Java virtual machine.

12. The byte code modification method of claim 1 wherein said first additional byte code instruction is to invoke a second method that refers to a dictionary, said dictionary having an entry for said method that identifies the location of a plug-in module containing a handler method that performs said first output function.

13. The byte code modification method of claim 12 wherein said method's byte code instructions are capable of being interpreted by a Java virtual machine, and, said first additional byte code instruction is an invokestatic instruction.

14. The byte code modification method of claim 12 wherein said method's byte code instructions are capable of being interpreted by a Java virtual machine, and, said first additional byte code instruction is an invokevirtual instruction.

15. The byte code modification method of claim 12 wherein said method's byte code instructions are capable of being interpreted by a Java virtual machine, and, said first additional byte code instruction is an invokespecial instruction.

16. The byte code modification method of claim 12 wherein said second additional byte code instruction is to invoke a third method that refers to said dictionary, said dictionary having an entry for said method that identifies the location of a plug-in module containing a handler method that performs said second output function.

17. The byte code modification method of claim 16 wherein said handler method that performs said first output function and said handler method that perform said second output function are the same handler method.
18. The byte code modification method of claim 1 wherein said modifying further comprises inserting a third additional byte code instruction, said third additional byte code instruction to cause a third output function to be executed for said method as a consequence of an error arising during execution of said method.
19. The byte code modification method of claim 1 wherein said modifying further comprises inserting an additional byte code instruction for each of said method's exit points to cause a second output function to be executed for said method as a consequence of any of said method's exit points being reached.
20. A byte code modification and distributed statistical recording method, comprising:

modifying a method's byte code instructions, said method comprising an entry point and an exit point, said modifying comprising inserting first and second additional byte code instructions into said method's byte code instructions, said first additional byte code instruction to cause a first output function to be executed for said method as a consequence of said entry point

being reached during runtime, said second additional byte code instruction to cause a second output function to be executed for said method as a consequence of said exit point being reached during runtime;

executing said method during runtime so as to execute said first and second output functions, said executing of said first and second output functions causing information concerning said method to be registered; and,

translating said information to a format employed within a distributed statistical records ("DSR") system.

21. The byte code modification and distributed statistical recording method of claim 20 wherein said first output function records a time of said entry point being reached.

22. The byte code modification and distributed statistical recording method of claim 21 wherein said first output function records input parameters provided to said method.

23. The byte code modification and distributed statistical recording method of claim 20 wherein said first output function records input parameters provided to said method.

24. The byte code modification and distributed statistical recording method of claim 20 wherein said second output function records a time of said exit point being reached.

25. The byte code modification and distributed statistical recording method of claim 24 wherein said second output function records output parameters provided by said method.

26. The byte code modification and distributed statistical recording method of claim 20 wherein said second output function records output parameters provided by said method.

27. The byte code modification and distributed statistical recording method of claim 20 wherein said first output function increments a counter maintained for said method.

28. The byte code modification and distributed statistical recording method of claim 20 wherein said second output function increments a counter maintained for said method.

29. The byte code modification and distributed statistical recording method of claim 20 further comprising compiling source code prior to said modifying to produce said method's unmodified byte code instructions.

30. The byte code modification and distributed statistical recording method of claim 20 wherein said byte code instructions are capable of being interpreted by a Java virtual machine.
31. The byte code modification and distributed statistical recording method of claim 20 wherein said first additional byte code instruction is to invoke a second method that refers to a dictionary, said dictionary having an entry for said method that identifies the location of a plug-in module containing a handler method that performs said first output function.
32. The byte code modification and distributed statistical recording method of claim 31 wherein said method's byte code instructions are capable of being interpreted by a Java virtual machine, and, said first additional byte code instruction is an invokestatic instruction.
33. The byte code modification and distributed statistical recording method of claim 31 wherein said method's byte code instructions are capable of being interpreted by a Java virtual machine, and, said first additional byte code instruction is an invokevirtual instruction.
34. The byte code modification and distributed statistical recording method of claim 31 wherein said method's byte code instructions are capable of being

interpreted by a Java virtual machine, and, said first additional byte code instruction is an invokespecial instruction.

35. The byte code modification and distributed statistical recording method of claim 31 wherein said second additional byte code instruction is to invoke a third method that refers to said dictionary, said dictionary having an entry for said method that identifies the location of a plug-in module containing a handler method that performs said second output function.

36. The byte code modification and distributed statistical recording method of claim 35 wherein said handler method that performs said first output function and said handler method that perform said second output function are the same handler method.

37. The byte code modification and distributed statistical recording method of claim 20 wherein said modifying further comprises inserting a third additional byte code instruction, said third additional byte code instruction to cause a third output function to be executed for said method as a consequence of an error arising during execution of said method.

38. The byte code modification and distributed statistical recording method of claim 20 wherein said modifying further comprises inserting an additional byte code instruction for each of said method's exit points to cause a second output

function to be executed for said method as a consequence of any of said method's exit points being reached.

39. A machine readable medium containing instructions which when executed cause a byte code modification method to be performed, the byte code modification method comprising:

modifying a method's byte code instructions, said method comprising an entry point and an exit point, said modifying further comprising inserting first and second additional byte code instructions into said method's byte code instructions, said first additional byte code instruction to cause a first output function to be executed for said method as a consequence of said entry point being reached during runtime, said second additional byte code instruction to cause a second output function to be executed for said method as a consequence of said exit point being reached during runtime.

40. The machine readable medium of claim 39 wherein said first output function records a time of said entry point being reached.

41. The machine readable medium of claim 40 wherein said first output function records input parameters provided to said method.

42. The machine readable medium of claim 39 wherein said first output function records input parameters provided to said method.

43. The machine readable medium of claim 39 wherein said second output function records a time of said exit point being reached.

44. The machine readable medium of claim 43 wherein said second output function records output parameters provided by said method.

45. The machine readable medium of claim 39 wherein said second output function records output parameters provided by said method.

46. The machine readable medium of claim 39 wherein said first output function increments a counter maintained for said method.

47. The machine readable medium of claim 39 wherein said second output function increments a counter maintained for said method.

48. The machine readable medium of claim 39 wherein the byte code modification method further comprises compiling source code prior to said modifying to produce said method's unmodified byte code instructions.

49. The machine readable medium of claim 39 wherein said byte code instructions are capable of being interpreted by a Java virtual machine.

50. The machine readable medium of claim 39 wherein said first additional byte code instruction is to invoke a second method that refers to a dictionary, said dictionary having an entry for said method that identifies the location of a plug-in module containing a handler method that performs said first output function.

51. The machine readable medium of claim 50 wherein said method's byte code instructions are capable of being interpreted by a Java virtual machine, and, said first additional byte code instruction is an invokestatic instruction.

52. The machine readable medium of claim 50 wherein said method's byte code instructions are capable of being interpreted by a Java virtual machine, and, said first additional byte code instruction is an invokevirtual instruction.

53. The machine readable medium of claim 50 wherein said method's byte code instructions are capable of being interpreted by a Java virtual machine, and, said first additional byte code instruction is an invokespecial instruction.

54. The machine readable medium of claim 50 wherein said second additional byte code instruction is to invoke a third method that refers to said dictionary, said dictionary having an entry for said method that identifies the location of a

plug-in module containing a handler method that performs said second output function.

55. The machine readable medium of claim 54 wherein said handler method that performs said first output function and said handler method that performs said second output function are the same handler method.

56. The machine readable medium of claim 39 wherein said modifying further comprises inserting a third additional byte code instruction, said third additional byte code instruction to cause a third output function to be executed for said method as a consequence of an error arising during execution of said method.

57. The machine readable medium of claim 39 wherein said modifying further comprises inserting an additional byte code instruction for each of said method's exit points to cause a second output function to be executed for said method as a consequence of any of said method's exit points being reached.